METHOD AND APPARATUS FOR PROVIDING DOMAIN CONVERSIONS FOR MULTIPLE CHANNELS AND APPLICATIONS THEREOF

RELATED PATENT APPLICATIONS

METHOD AND APPARATUS FOR PROVIDING DATA FOR SAMPLE RATE CONVERSION having an attorney docket number of SIG000063 and a filing the date the same as the present patent application;   and

METHOD AND APPARATUS FOR ADJUSTING TIMING IN A DIGITAL SYSTEM having an attorney docket number of SIG000060 and a filing the date the same as the present patent application.

TECHNICAL FIELD OF THE INVENTION

This invention relates generally to telecommunications and more particularly to an analog front-end for use in such telecommunication systems.

BACKGROUND OF THE INVENTION

As is known, data may be communicated from one entity (e.g. end users, computers, server, facsimile machine et cetera) to another entity via a communication infrastructure. The communication infrastructure may include a public switch telephone network (PSTN), the Internet, wireless communication system, and/or a combination thereof. Such a communication infrastructure supports many data communication protocols, which prescribe

the formatting of data for accurate transmission from one
entity to another. Such data communication protocols
include digital subscriber line (DSL), asymmetrical digital
subscriber line (ADSL), universal asymmetrical digital

5   subscriber line (UADSL or G.Lite), high-speed digital
subscriber line (HDSL), symmetrical high-speed digital
subscriber lines (HDSL), asynchronous transfer mode (ATM),
Internet protocol (IP), et cetera.

10   Each of the various data transmission protocols
prescribes the formatting of data into frames. Each frame
may include a header section, which identifies information
particular to the frame, and a data section, which carries
the communication data. The data section may be divided

15   into a plurality of data segments, time slots, carrier-
frequency bins, packets, et cetera. Depending on the
particular data transmission protocol, a frame of data will
be transmitted in a continuous manner or in a discontinuous
manner. For example, IP and ATM data transmission

20   protocols packetize a frame of data and the packets are
transmitted in a discontinuous manner. In contrast, xDSL
data transmission protocols require the frames to be
transmitted in a continuous manner.

25   For xDSL data transmission protocols, the data is
processed within a modem of a given entity in the digital
domain and converted to the analog domain for transmission
via the communication infrastructure. Conversely, data is
received via the communication infrastructure in the analog

30   domain and converted into the digital domain for further
processing. For xDSL modems, the analog to digital
conversion and digital to analog conversion are done in an

analog front-end. As the need for further integration and functionality of modems increases, the need for more complex analog front-ends increases accordingly.

5   Therefore, a need exists for a method and apparatus that provides domain conversion for multiple channels, e.g. telecommunication paths.

BRIEF DESCRIPTION OF THE DRAWINGS

10

Figure 1 illustrates a schematic block diagram of a multi-channel analog front-end in accordance with the present invention;

15   Figure 2 illustrates an alternate multi-channel front-end in accordance with the present invention;

Figure 3 illustrates a schematic block diagram of another alternate multi-channel analog front-end in 20   accordance with the present invention;

Figure 4 illustrates a schematic block diagram of yet another multi-channel analog front-end in accordance with the present invention;

25

Figure 5 illustrates a schematic block diagram of an apparatus for providing domain conversions for multiple channels in accordance with the present invention; and

30   Figure 6 illustrates a logic diagram of a method for providing domain conversions for multiple channels in accordance with the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Generally, the present invention provides a method and
apparatus for domain conversions for multiple channels
within a single analog front-end. The method and apparatus
include processing that begins by generating a system
clock. The processing continues by converting a frequency
of 1st data from a 1st channel frequency to a 2nd frequency
based on a 1st integer ratio of the system clock. For
example, the data rate of the data in a 1st channel is
converted to a system level rate. The processing continues
by converting the domain of the 1st data rate from a 1st
domain to a 2nd domain. For example, the domain is
converted from the analog domain to the digital domain or
the digital domain to the analog domain. The processing
continues by converting a frequency of the 2nd data of a 2nd
channel from a 2nd channel frequency to the 2nd frequency
based on a 2nd integer ratio of the system clock. For
example, the rate of the 2nd data may be different than the
rate of the 1st but both are converted to the 2nd frequency,
which is universally used within the analog front-end. The
processing continues by converting the domain of the 2nd
data from the 1st domain to the 2nd domain. With such a
method and apparatus, a multiple channel analog front-end
is achieved in an integrated format that provides for
increased integration of telecommunication services and a
corresponding decrease in cost per telecommunication
services.

The present invention can be more fully described with
reference to Figures 1 through 6. Figure 1 illustrates a

schematic block diagram of a multi-channel analog front-end
10 that includes a 1st channel path 12, a 2nd channel path 14
and a system clock module 16. The system clock module 16
is operably coupled to a crystal 18 and produces a system

5   clock 20 that has a given frequency, for example 35 MHz.
Depending on the desired frequency, the system clock module
16 may include a simple pair of inverters, a pair of
inverters and a phase lock loop, and/or any known mechanism
for generating a reliable clock signal from a crystal.

10   Note that the crystal 18 may be eliminated if the system
clock 20 is available in the final application from another
clock source.

The 1st channel path 12 includes a 1st sample rate

15   converter 22 and a 1st domain conversion module 24. The 1st
sample rate converter 22 receives the 1st data at a 1st
channel frequency in a 1st domain. Based on a 1st integer
ratio 26, which is a ratio based on the 1st channel
frequency and the system clock 20, the sample rate

20   converter 22 produces the 1st data at a 2nd frequency in the
1st domain. For example, the 1st domain may be the digital
domain such that the 1st sample rate converter 22 converts
the rate of the digital 1st data from the data rate of the
first channel path to a second frequency, which is used

25   throughout the analog front-end 10. The 1st domain
conversion module 24 receives the 1st data at the 2nd
frequency in the 1st domain and converts it into the 1st data
at the 2nd frequency in the 2nd domain. For example, the 1st
domain conversion module 24 converts the data from the

30   digital domain to the analog domain.

The 2nd channel path 14 includes a 2nd sample rate converter 28 and a 2nd domain conversion module 30. The 2nd sample rate converter 28 is operably coupled to receive 2nd data at a 2nd channel frequency in the 1st domain and to

5 produce the 2nd data at a 2nd frequency in the 1st domain based on the 2nd integer ratio 32, which is a ratio between the system clock 20 and the 2nd channel frequency. As such, the sample rate converter 28 converts the frequency of the 2nd data from the 2nd channel frequency rate to the 2nd

10 frequency. The 2nd domain conversion module 30 receives the 2nd data at the 2nd frequency in the 1st domain and converts it into the 2nd data at a 2nd frequency in the 2nd domain.

As one of average skill in the art will appreciate,

15 the 1st domain conversion module 24 and the 2nd domain conversion module 30 may receive their respective data at the 2nd frequency in the 2nd domain and convert it into the respective data at the 2nd frequency into the 1st domain. Similarly, the 1st sample rate converter 22 and the 2nd

20 sample rate converter 28 may receive the 1st and 2nd data at the 2nd frequency and convert it into the 1st and 2nd data at their respective channel frequencies. As one of average skill in the art will further appreciate, the 1st data corresponds to a 1st channel of a multi-channel

25 telecommunication system and the 2nd data corresponds to the 2nd channel in the telecommunication system. As such, multiple communications may be supported via a single analog front-end as shown in Figure 1. Conversely, multi-channels may be integrated into a single communication and

30 supported by the analog front-end of Figure 1.

Figure 2 illustrates a schematic block diagram of an alternate multi-channel analog front-end 40. The analog front-end 40 includes the system clock module 16, the 1st channel path 12, the 2nd channel path 14, a controller 42, a 3rd channel path 46, and a 4th channel path 52. The 1st and 2nd channel paths 12 and 14 include similar components and function in a similar manner as described with reference to Figure 1. The controller 42 is operably coupled to receive the system clock 20 and produce integer ratios 44. The integer ratios 44 include the 1st integer ratio 26, the 2nd integer ratio 32, a 3rd integer ratio 47, and a 4th integer ratio 53. The controller, or control module, 42 determines the integer ratios based on the desired frequencies of the respective channel paths and the system clock. As such, the controller 42 may include circuitry to determine the frequency differences between the system clock and the frequencies of the channel paths to produce the integer ratios, it may include a lookup table based on the known data rates of the channel paths, or it may include a series of registers that are programmable by the user of the system.

The 3rd channel path 46 includes a 3rd sample rate converter 48 and a 3rd domain conversion module 50. The 3rd domain conversion module 50 receives 3rd data at the 2nd frequency in the 2nd domain and converts it into the 3rd data at the 2nd frequency in the 1st domain. The 3rd sample rate converter 48 receives the 3rd data at the 2nd frequency in the 1st domain and converts it into 3rd data at the 3rd channel frequency in the 1st domain.

The 4th channel path 52 includes a 4th sample rate
converter 54 and a 4th domain conversion module 56. The 4th
domain conversion module 56 receives 4th data at the 2nd
frequency in the 2nd domain and converts it into the 4th data

5 at the 2nd frequency in the 1st domain. The 4th sampling
rate 54, based on the 4th integer ratio 53, produces the 4th
data at the 4th channel frequency in the 1st domain.

As one of average skill in the art will appreciate,

10 the 1st and 2nd domain conversion modules may be digital to
analog converters while the 3rd and 4th domain conversion
modules 50 and 56 may be analog to digital converters. As
such, the 1st and 3rd paths may be used for transmit and
receive paths of one telecommunication channel while the 2nd

15 and 4th paths may be used for transmit and receive paths of
a 2nd telecommunication channel.

Figure 3 illustrates a schematic block diagram of an
analog front-end 60 that includes the 1st sample rate

20 converter 22, the 1st domain conversion module 24, the 2nd
sample rate converter 28 and the 2nd domain conversion
module 30. The 1st sample rate converter 22 is shown to
include a receiver module 62, which is operably coupled to
receive a word of the 1st data. The word may be a single

25 bit, multiple bits, 8 bits, 16 bits, et cetera of the 1st
data. The receiver module 62 stores the word of the 1st
data as a stored word 72. The rate conversion module 64,
which may perform an integer rate conversion 66 based on
the 1st integer ratio 26, receives the stored word 72 and

30 produces a sample rate converted word 74. The integer rate
conversion 66 may be a function that repeats the word based
on the 1st integer ratio 26. For example, if the 1st integer

ratio is 3, the sample rate converted word 74 is 3 replications of the stored word 72. As such, the 1st sampling rate converter 22, for this example, changes the rate of the 1st data by a factor of 3.

5

The 1st domain conversion module 24 includes a digital to analog converter 68 and a filter 70. The digital to analog converter 68 is operably coupled to receive the sample rate converted word 74 and convert it into an analog

10   signal. The analog signal is filtered by filter 70 to produce the 1st data at the 2nd frequency in the 2nd domain, which for this path is the analog domain.

The 2nd sample rate converter 28 includes a receiver

15   module 76, and a rate conversion module 78, which may perform the integer rate conversion 66. The receiver module 76 is operably coupled to receive a word of the 2nd data and produce therefrom a stored word 84. The rate conversion module 78 based on the 2nd integer ratio 32

20   produces a sample rate converted word 86 from the stored word 84.

The 2nd domain conversion module 30 includes a digital to analog converter 80 and a filter 82. The digital to

25   analog converter 80 receives the sample rate converted word 86 and produces an analog representation thereof. The analog signal is filtered via filter 82, which outputs the 2nd data at the 2nd frequency in the 2nd domain. Note that the implementation of sample rate converter 22 and 28 could

30   include any number of known sample rate conversion algorithms such as linear interpolation, table look up, etc.

Figure 4 illustrates a schematic block diagram of an analog front-end 90 that includes the 1st sample rate converter 22, the 1st domain conversion module 24, the 2nd

5　sample rate converter 28, and the 2nd domain conversion module 30. The 1st domain conversion module 24 includes an analog to digital converter 94 and a filter and pre-amp circuit 92. The filter and pre-amp circuit 92 is operably coupled to receive the 1st data at the 2nd frequency in the

10　2nd domain (e.g. the analog domain), filter it, and amplify it. The filter and pre-amplifier 92 provides the filtered and amplified version of the 1st data to the analog to digital converter 94. The analog to digital converter 94 converts the domain of the 1st data into the digital domain

15　thereby producing the 1st data at the 2nd frequency in the 1st domain.

The 1st sample rate converter 22 includes a receiver module 96 and a rate conversion module 98, which may do an

20　interpolative rate conversion 100. The receiver module 96 receives a word of the 1st data at the 2nd frequency in the 1st domain and stores it to produce a stored word 102. The rate conversion module 98 receives the stored word 102 and, based on the 1st integer ratio 26, produces a word, partial

25　word, or multiple words of the 1st data at the 1st channel frequency in the 1st domain by utilizing an interpolative rate conversion process 100. The interpolative process utilizes linear or higher order functions of words to produce the resulting word of the 1st data at the 1st channel

30　frequency in the 1st domain. For example, if the integer ratio is 2.5, the 1st two words would be replications of the stored word while the 3rd word would be an interpolated word

based on the words between the 1st stored word and a 2nd
stored word. Alternatively, the rate conversion could be
implemented as a decimation with the channel frequency less
than or equal to the second frequency.

5

The 2nd domain conversion module 30 includes an analog
to digital converter 106 and a filter and pre-amp circuit
104. The filter and pre-amp circuit 104 is operably
coupled to receive 2nd data at the 2nd frequency in the 2nd

10   domain (e.g. analog domain). The filter and pre-amp
circuit process the 2nd data and provides the processed data
to the analog to digital converter 106. The analog to
digital converter converts the 2nd data into digital data,
which is designated the 2nd data at the 2nd frequency in the

15   1st domain.

The 2nd sample rate converter 28 includes a receiver
module 108 and a rate conversion module 110, which may
perform an interpolative rate conversion 112. The receiver

20   module 108 receives a word at a time of the 2nd data at the
2nd frequency in the 1st domain to produce a stored word 114.
The rate conversion module 110 receives the stored word 114
and, based on the 2nd integer ratio 32, produces the 2nd data
at the 2nd channel frequency in the 1st domain. The rate

25   conversion module 110 may utilize an interpolative rate
conversion process 112, or a decimation rate conversion
process (not shown), to perform the sample rate conversion.

As one of average skill in the art will appreciate,

30   the rate conversion modules shown in Figures 3 and 4 may
utilize an interpolative rate conversion, decimation sample
rate conversion, an integer rate conversion, and/or any

other known mechanism for sample rate conversion of data. As one of average skill in the art will also appreciate, by converting the channel frequencies all into the 2nd frequency, all analog to digital conversion and digital to

5 analog conversion is done on data having the same frequency. As such, noise, jitter, et cetera, which may be produced by having multiple clocks on a single substrate are substantially eliminated in the present analog front-ends.

10

Figure 5 illustrates a schematic block diagram of an apparatus 120 for providing domain conversions for multiple channels. The apparatus includes a processing module 122 and memory 124. The processing module 122 may be a single

15 processing device or a plurality of processing devices. Such a processing device may be a microprocessor, microcontroller, digital signal processor, central processing unit, state machine, logic circuitry, and/or any device that manipulates signals (analog or digital) based

20 on operational instructions. The memory 124 may be a single memory device or a plurality of memory devices. Such a memory device may be read-only memory, random access memory, floppy disk memory, system memory, hard drive memory, and/or any device that stores digital information.

25 Note that when the processing module 122 implements one or more of its functions via a state machine or logic circuitry, the memory storing the corresponding operational instructions is embedded within the circuitry comprising the state machine or logic circuit. The operational

30 instructions stored in memory 124 and processed by the processing module 122 are described in a logic diagram as shown in Figure 6.

Figure 6 illustrates a logic diagram of a method for providing domain conversions of multiple telecommunication channels on a single analog front end. The process begins at step 130, where a system clock is generated. The process then proceeds to steps 132, 136, 140, and 146. At step 134, a frequency of the 1st data is converted from a 1st channel frequency to a 2nd frequency based on a 1st integer ratio of the system clock. The 1st integer ratio of the system clock is determined based on the 2nd frequency, i.e. the frequency at which all channels are converted to, and the frequency of the 1st data. The conversion may be done by storing a word of the 1st data to produce a stored word. The stored word is then replicated based on the 1st integer ratio to produce a sample rate converted word, which may be done by utilizing an integer replication or an interpolation replication.

The process then proceeds to Step 134 where the domain of the 1st data is converted from a 1st domain to a 2nd domain. Note that the 1st domain may be a digital domain and the 2nd domain may be an analog domain. As such, the transmit data in a communication system may have its rate converted via a sample rate converter and then its domain converted by a digital to analog converter.

At Step 136, a frequency of 2nd data is converted from a 2nd channel frequency to the 2nd frequency, i.e. the desired operating frequency of the integrated circuit, based on a 2nd integer ratio of the system clock. The 2nd integer ratio may be determined in a similar manner as the 1st integer ratio. The process then proceeds to Step 138

where the domain of the 2nd data is converted from the 1st domain to the 2nd domain.

At Step 140, the domain of the 3rd data is converted from the 2nd domain to the 1st domain. The 2nd domain may be an analog domain and the 1st domain may be the digital domain. The process then proceeds to Step 140, where a frequency of 3rd data is converted from the 2nd frequency to a 3rd channel frequency based on a 3rd integer ratio of the system clock.

At Step 144, the domain of the 4th data is converted from the 2nd domain to the 1st domain.. The process then proceeds to Step 146 where a frequency of the 4th data is converted from the 2nd frequency to a 4th channel frequency based on a 4th integer ratio of the system clock.

The preceding discussion has presented a method and apparatus for providing domain conversions for multiple channels in a single integrated analog front-end. As one of average skill in the art will appreciate, other embodiments may be derived from the teachings of the present invention without deviating from the scope of the claims. For example, more than two analog to digital converters and digital to analog converters may be utilized on a single integrated analog front-end, where each analog to digital converter and digital to analog converter pair constitute a transmit and receive telecommunication channel.